Library Management Analysis - Sorting Algorithms

# 1. Linear Search

Linear search is a simple search algorithm that checks each element in the list one by one until the desired element is found or the end of the list is reached.

## Time Complexity

- Best Case: O(1) (element found at the beginning)  
- Average Case: O(n)  
- Worst Case: O(n)

## Use Cases

- Works on unsorted data  
- Suitable for small data sets  
- Easy to implement

## Pros

- Simple to implement  
- No need to sort the data

## Cons

- Inefficient for large data sets  
- Slower compared to more advanced algorithms

# 2. Binary Search

Binary search is an efficient algorithm for finding an element in a sorted list by repeatedly dividing the search interval in half.

## Time Complexity

- Best Case: O(1) (element found at the middle)  
- Average Case: O(log n)  
- Worst Case: O(log n)

## Use Cases

- Works only on sorted data  
- Best suited for large, sorted data sets

## Pros

- Much faster than linear search for large sorted datasets  
- Reduces search time significantly

## Cons

- Requires sorted data  
- Slightly more complex to implement